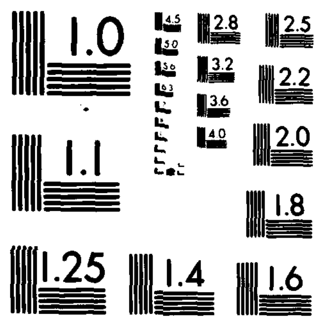
AD-A081 783 WHARTON SCHOOL PHILADELPHIA PA DEPT OF DECISION SCIENCES F/G 9/2
RATIONAL DATA BASE STANDARDS: AN EXAMINATION OF THE 1978 CODASY--ETC(U)
OCT 78 E K CLEMONS N00014-75-C-0462
UNCLASSIFIED 78-10-02 NL

END
DATE
FILMED
4 80
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

sion Scie

LEVEL II

RATIONAL
DATA BASE STANDARDS:
AN EXAMINATION OF THE
1978 CODASYL DDLC REPORT

ERIC K. CLEMONS

78-10-02

RATIONAL DATA BASE STANDARDS:

AN EXAMINATION OF THE
1978 CODASYL DDLC REPORT

Eric K. Clemons

Working Paper 78-10-02

1 October 1978

Department of Decision Sciences
The Wharton School
University of Pennsylvania
Philadelphia, PA 19104

Research supported in part by Office of Naval Research Contract N00014-75-C-0462

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 14 78-10-02	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) 6 RATIONAL DATA BASE STANDARDS: AN EXAMINATION OF THE 1978 CODASYL DDLC REPORT.		5. TYPE OF REPORT & PERIOD COVERED Technical Report.
		6. PERFORMING ORG. REPORT NUMBER 78-10-02
7. AUTHOR(s) 10 Eric K./Clemons		8. CONTRACT OR GRANT NUMBER(s) 15 N00014-75-C-0462
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Decision Sciences The Wharton School Philadelphia, PA 19104		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task NR049-272
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy 800 N. Quincy St., Arlington, VA 22217		12. REPORT DATE 11 1 Oct[redacted]78
		13. NUMBER OF PAGES 12 (12) 17
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

CODASYL, ANSI/SPARC three-schema architecture, data base, DDLC (Data Description Language Committee), schema and subschema facility

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The CODASYL DDLC 1978 Report incorporates numerous enhancements and language changes. Unfortunately, the major design limitations associated with earlier reports and specifications, in particular a schema facility too closely related to machine rather than enterprise requirements and an extremely limited subschema facility, are retained. We suggest that the recent CODASYL specifications remain inappropriate as either an instance of an ANSI/SPARC three-schema architecture or as a candidate for a national data base system standard.

408 7__

JLG

RATIONAL DATA BASE STANDARDS: AN EXAMINATION OF THE 1978 CODASYL DDLC REPORT

ABSTRACT

The CODASYL Data Description Language Committee's 1978 Report incorporates numerous enhancements and language changes made since the earlier 1971 and 1973 reports. Unfortunately, the major design limitations associated with these earlier specifications, in particular a schema facility too closely related to machine rather than enterprise requirements and an extremely limited subschema facility, are retained.

After examination of these limitations, we suggest that the recent CODASYL specifications remain inappropriate as either an instance of an ANSI/SPARC three-schema architecture or as a candidate for a national data base system standard. A long term strategy for the development of a more rational proposal for standardization is suggested. And a short term strategy is offered, one that permits rational planning for and implementation of data base conversions to occur today, without concern that subsequently developed standards might render obsolete the conversion effort and data base management system selected.

I. INTRODUCTION

We are addressing two related questions:

1. What is the suitability of the CODASYL 1978 DDL specifications [13] as a candidate for adoption as a national data base system standard?

2. Do these specifications match well with those of the 1975 [1] and 1977 [23] ANSI/X3/SPARC proposals for a three-schema data base architecture?

I think that many arguments in favor of rapid agreement on a data base standard are clear. Every organization has a large investment in data and data processing software; there is pressure on management to convert to a data base architecture, converting existing data and programs to realize the savings and additional benefits believed to accrue from an integrated data base management system; and it is crucial that the considerable expense associated with this conversion not be wasted by subsequent agreement on a standard that renders obsolete the data base system chosen [4]. Likewise, as users wish to avoid the expenses of unnecessary data base conversions, so too do implementors and vendors of data base systems wish to avoid unnecessary modifications and alterations of their products. Indeed, since the 1978 CODASYL specifications differ significantly from earlier specifications [19], there is a certain reluctance on the part of some implementors to modify their systems to meet these new specifications, because there is no guarantee that they will remain fixed for a period sufficient to recover conversion costs.

Systems conforming to CODASYL specifications have been chosen by many corporate users; likewise, CODASYL is the only model with sufficient vendor support to be considered as a serious candidate for a standard. In fact, the CODASYL specifications are rapidly emerging as a de facto American data base system standard. I feel very strongly that this is unfortunate; the CODASYL model, in its present form, is largely inappropriate.

Fortunately, there exists an alternative to the premature adoption of a standard: It is only necessary to decide on a "kernel" of a standard, a component of the programmer interface that will be supported in any future data base standard. Here, the CODASYL model fares somewhat better. It is in widespread use, making it a logical choice. And the ANSI/SPARC proposals which will no doubt have a major influence on future data base management system technology permit great flexibility in any subsequently adopted standards; thus the kernel may be only one of several, dramatically different interfaces supported. Also, the low level of the CODASYL data manipulation language and the limited inter-schema mapping facilities supported should make inclusion of a CODASYL interface relatively easy and inexpensive.

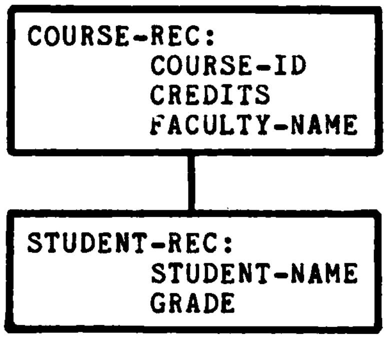
II. SHORTCOMINGS OF CODASYL SPECIFICATIONS

My principal objection to the CODASYL system is its lack of concern for and support of the programming user. This is not an objection to the design, level, or syntax of the current DML -- if so it would be only a superficial objection -- rather, it is an objection to the form of subschema provided.

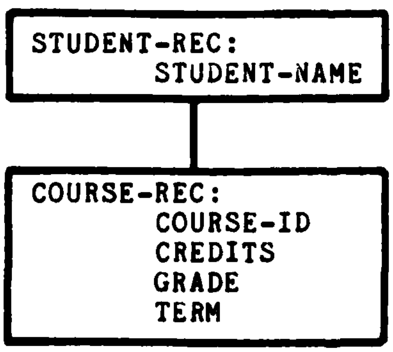
The CODASYL system is not appropriate as an instance of the ANSI/SPARC three-schema architecture. It pre-dates the ANSI/SPARC proposal and does not successfully capture its philosophy. While the 1978 DDL specifications include a proposal for a new data storage description language (DSDL) and thus include three schemas, they are not the correct three schemas: The DDL schema is not purely conceptual, but contains constructs better placed in the internal schema as they deal primarily with access efficiency [10]. The subschema facility is even farther from an external schema facility, including both conceptual and internal level constructs. The resulting design is not clean and does not provide adequate separation of functions; this is significant, not because ANSI/SPARC proposal represents an absolute standard that must be closely followed, but because the limitations of the selected CODASYL design have unfortunate implications for programming ease and programmer productivity, data independence, and distributed processing.

Likewise, I feel that the CODASYL system is not appropriate for adoption as a national data base standard, again because of limitations of the subschema facility and the programming interface. In order to understand the orientation and limitations of the system, it is necessary to remember the period -- late 1960s -- in which its original design and specification were prepared. The principal concerns of the Data Base Task Group were to provide a limited increase in flexibility and generality of data base systems without incurring substantial penalties in reduced machine efficiency. Thus, networks of associated records provide greater generality than simple hierarchies; by freezing the supported associations to be those explicitly declared in sets, flexibility is limited but efficient access is assured. Similarly, by limiting maps between schema and subschemas to a few simple forms, efficient operation is preserved. Unfortunately, the resulting design, while efficient, is too limited; in several ways it is inappropriate for the technology and demands of contemporary data processing, a decade later and in the future.

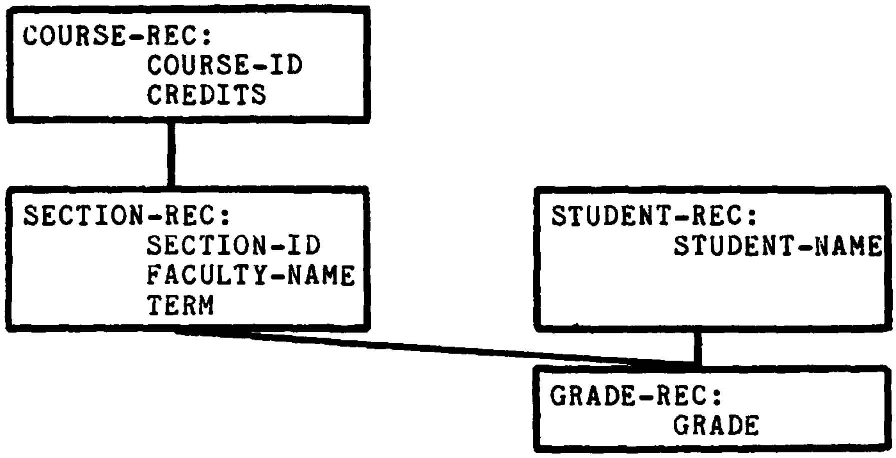
These limitations stem, principally, from the fact that the subschema follows the schema too closely in form. Individual records in the schema map to single records in the subschema, and data associations remain by set membership. In general, networks exist in a data base not because any single user requires so general a structure, but because the collection of hierarchical associations required by each user are incompatible [7]. Thus, if one user wants a hierarchical association between courses he taught and all student grades for the courses:



while another user wants a hierarchical association between a student and all course grades received:



this will probably be captured at the conceptual level with a network of the following form:



At the external or subschema level users should not see networks but rather the hierarchies required for their individual applications. In fact, where possible the details of the conceptual schema, its record types and set associations, should be hidden from the user. Navigation, data association made using DML statements exploiting set membership, is only slightly removed from manipulation using record keys or device addresses. Such navigation should not be necessary. Rather, subschema records should be in direct correspondence, not with schema records, but with the cognitive structures used by programmers in the solving of problems and the design of algorithms. Thus a STUDENT-TRANSCRIPT subschema record would be a single record comprising student name and a repeating group containing course, grade, and term data; the user would request this record with a single DML statement, although it may correspond to dozens of schema records, of four record types, linked by membership in three sets.

The design limitations of the CODASYL subschema facility have undeniable implications for the process of application program development, maintenance, and execution.

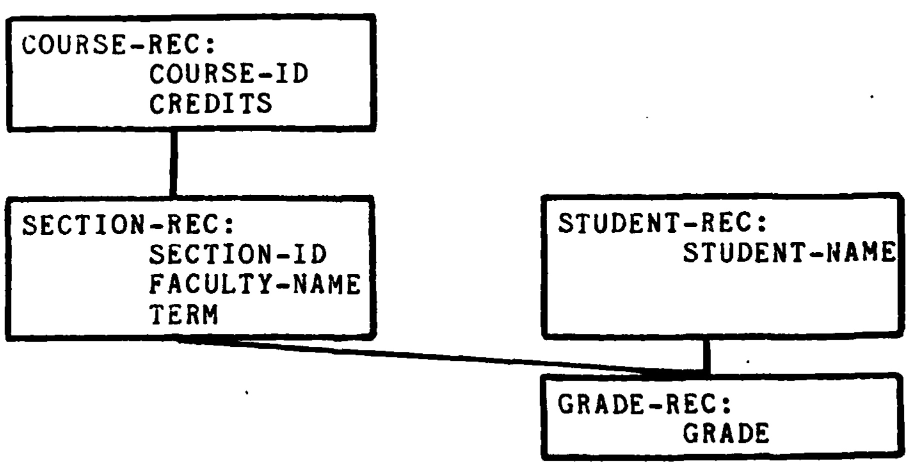
1. Because the subschema structures are in close correspondence, not with user cognitive structures, but with structures provided for the complete enterprise data model, considerable user navigation is required to make necessary data associations and to construct the relevant information objects. This process is difficult, slow, and prone to error; obviously programmer productivity is affected.

2. In the CODASYL model, changes or extensions to the set of supported applications may well result in major structural changes to the schema; e.g., addition of a new application may change a schema

hierarchy to a confluency. Because of the close correspondence between schema and subschema records, the application programs are not buffered from this change, and thus may require major redesign and reprogramming effort. Moreover, the semantics of existing data associations, made by DML accesses and host language iteration and qualification, are very difficult to determine from the programs. Redesign will not be an easy, automated process; rather it will be manual and difficult. Obviously, data independence is affected [21].

3. Again, because of the level of CODASYL DML and the close relationship between schema and subschema, a number of data selection procedures (e.g., ignore records with the following data values) and data reduction procedures (e.g., return only average balances, grouped by class and status of account) are performed by the application programs. Specified in the schema to subschema map, these procedures could be performed by a "data base machine" supporting the DBMS, rather than by the user program, substantially reducing the volume of data actually returned to the user program. Thus, channel traffic and communications expenses in a distributed environment are affected.

To make concrete the terms and objections stated, we consider as an example a data base again containing student course information. In the schema we have student records related to grade, course, and section grades as follows:



From this we want to construct a summary transcript, with student name, average grade point, and average grade point for each term:

```
01  SUMMARY-TRANSCRIPT.
    02  STUDENT-NAME ... .
    02  GRADE-POINT ... .
    02  TERM-ENTRY OCCURS ...
        03  TERM-ID ... .
        03  TERM-AVERAGE ... .
```

With an external schema facility, retrieval of this transcript is requested with a single READ; changes to the conceptual schema structure that change record types and associations alter inter-schema mapping functions but not application programs; and in a distributed environment the data base machine can transmit the desired summaries, rather than the grade and course credit and term information needed to compute these summaries. Also, we note that employing the current DML to compute these summaries, the user must:

1. FIND all GRADE records for a student

2. for each GRADE, FIND and GET the owner SECTION record

3. sort SECTION records in ascending order by term

4. make each SECTION record current, in order by term

5. for each SECTION record, as it becomes current, FIND and GET the owner COURSE record to get credit information. Also, for each current SECTION and the desired student, the member GRADE record must again have a FIND and GET to get the actual grade received.

6. with the information obtained in the preceding step, host language arithmetic statements are used to compute the desired averages.

Clearly, obtaining the information with a single READ is preferable.

III. AN ALTERNATIVE EXTERNAL SCHEMA FACILITY

It is of limited usefulness to criticize a system design, without proposing an alternative. As an alternative, I offer a greatly enhanced subschema facility, one that in effect offers each user a virtual data base with simple structure corresponding to the specific needs of each application program.

Such a facility has three basic requirements. To construct schema to subschema maps it is necessary to specify:

1. access information

2. restructuring information

3. data item definition

Access information specifies from which records data are to be obtained, what data values are necessary for qualification, and which set membership or other access paths are to be employed to make the necessary associations. Restructuring information controls repetition (e.g., the inclusion of all term summaries in a single summary transcript in the example of section II), grouping (e.g., grouping of grade information by the term that the course was taken), and whether complete content or summary only data are to be included (e.g., include only summary over-all average and term averages, but no individual course grades). Data item definition includes specifying the source of data items actually present in the schema, as well as rules for preparing virtual computed items and structured items. A detailed description of such a general external schema facility for a relational environment is available [7]; language enhancements for a CODASYL system are in preparation [11]. Such a facility will greatly simplify the programmer's interaction with data base systems, while leaving concern for enterprise support and machine efficiency to other schema levels, as is appropriate.

IV. A CANDIDATE FOR STANDARDIZATION?

I do not propose that any current research on external schema facilities be given serious study as a candidate for standardization at this time. Several technical problems remain, requiring technical study; likewise, several questions concerning human factors design and performance remain unanswered. An efficient implementation of a general external schema facility appears difficult; naive approaches suffer from explosive growth of required secondary storage and machine processing time. Equally important, the problem of data base update in a multi-schema environment remains unsolved: surprisingly few maps from conceptual schema to external schema are invertible, implying that for most user updates to data at the level of the user's virtual data base, corresponding changes to the stored data base cannot be determined [5, 8].

Perhaps the most important consideration in any language, interface, or architecture design is their effect on programmer performance, in particular programmer productivity and program correctness and ease of maintenance. There has been some interest in human factors study and some guidelines have been given [20]; some interesting experiments have been performed [16, 17, 22] but there has been no conclusive work produced.

I estimate that resolution of technical design problems and human factors questions is two or three years in the future; preparation of potential standards, based on this work, will require still more time.

V. WHAT DO WE DO NOW?

It is apparent that we cannot wait three to five years for the adoption of national standards, but must act now. Perhaps it is more accurate to say that if we do not act rapidly, we will have lost the potential for rational choice: sheer volume of existing implementations and in-progress conversions based on systems currently commercially available will dictate a standard.

Therefore, my suggestion made originally in section I appears reasonable: We should agree that any future standard for data base architecture must include the current CODASYL DML and subschema facility in its programmer interface, permitting data base conversions to be planned and performed now. We should also agree that, after five years, the facilities for CODASYL schema, subschema, and DSDL schema will be re-evaluated, based on advances in the areas of external, conceptual, and internal schema research. Perhaps, as a result of these advances, CODASYL specifications will have only limited resemblance to current specifications. Or, perhaps, future standards will preserve nothing of the current CODASYL specifications beyond that which is explicitly included in the kernel.

I believe that much additional research in the area of the conceptual schema is required. Recent work by Bachman and Daya [3], Chen [6], and Gerritsen and Lee [15] indicate the potential for representing data base semantics as well as structure in the schema,. Work on external schema facilities, based on my own research cited earlier and the implementation results of the IBM System R group [2] must continue, and must be subjected to human factors study and evaluation. Work by CODASYL at the internal schema level will continue. It is to be hoped that the results of these separate efforts can be combined, within the framework of an ANSI/SPARC three-schema architecture, to produce a data base architecture appropriate to the needs of business and government in the decade ahead.

ACKNOWLEDGEMENTS

I would like to acknowledge my debt to my colleagues at the University of Pennsylvania, Rob Gerritsen and Frank Germano, to my co-panelists at the Fourth Very Large Data Base Conference, John Berg, Frank Manola, and Diane C. P. Smith; and to my co-participants at the N.B.S. Three-Schema Feasibility Workshops, particularly Don Chamberlin, Henry Lefkowitz, and Carlo Zaniolo. Their thinking has influenced my thinking. Obviously, biases, errors, and omissions in this paper are my own.

REFERENCES

1. "ANSI/X3/SPARC Study Group on Data Base Management Systems Interim Report 75-02-08". FDT--Bulletin of the ACM SIGMOD, Vol. 7, No. 2, 1975.

2. Astrahan, M. M. et al. "System R: A Relational Approach to Database Management". ACM Transactions on Database Systems, Vol. 1, No. 2, 1976, pp. 97-137.

3. Bachman, C. W., and Daya, M. "The Role Concept in Data Models". Proceedings, Third International Conference on Very Large Data Bases, Tokyo, Japan, Oct. 1977, pp. 464-476.

4. Berg, J. L. "Implementing a Framework for DBMS Standards". Unnumbered Working Paper, Institute for Computer Sciences and Technology, National Bureau of Standards, Gaithersberg, MD, Mar. 1978.

5. Bernstein, P. A. and Dayal, U. "On the Updatability of Relational Views". Proceedings of the Fourth International Conference on Very Large Data Bases, West Berlin, Germany, Sept. 1978, pp. 368-377.

6. Chen P. P.-S. "The Entity-Relationship Model--Toward A Unified View of Data". ACM Transactiions on Database Systems, Vol. 1, No. 1, 1976, pp. 9-36.

7. Clemons, E. K. Design of a User Interface for a Relational Data Base, Dissertation, School of Operations Research, Cornell University, 1976.

8. Clemons, E. K. "An External Schema Facility to Support Data Base Update". Databases: Improving Usability and Responsiveness, ed. Ben Shneiderman, Academic Press, New York, 1978, pp. 371-398.

9. Clemons, E. K. "The External Schema and CODASYL". Proceedings, Fourth International Conference on Very Large Data Bases, Berlin, West Germany, Sept. 1978, p. 130.

10. Clemons, E. K. "The CODASYL 1978 DDL Specifications: A Critical Evaluation". Decisions Sciences Working Paper in Progress, The Wharton School, University of Pennsylvania, 1978.

11. Clemons, E. K. and Germano, F. "Design of an External Schema Facility for CODASYL Data Base Management Systems". Decision Sciences Working Paper in Progress, The Wharton School, University of Pennsylvania, 1978.

12. "CODASYL Data Base Task Group April 71 Report". ACM, New York, 1971.

13. "CODASYL Data Description Language Committee Journal of Development", 1978.

14. Gerritsen, R. "Conceptual and Internal Schemas in CODASYL", Proceedings, Fourth Internation Conference on Very Large Data Bases, Berlin, West Germany, Sept. 1978, p. 131.

15. Gerritsen, R. and Lee, R. "Extended Semantics for Generalization Hierarchies". Proceedings, ACM SIGMOD Workshop, Austin, Texas, June 1978, pp. 18-25.

16. Kuhn, M. and Shneiderman, B. "Two Experimental Comparisons of the Relational and Hierarchical Database Models". IFSM Technical Report No. 31, University of Maryland, Feb. 1978.

17. Lochovsky, F. and Tsichritzis, D. "User Performance Considerations in DBMS Selection", Proceedings, ACM SIGMOD Workshop, Toronto, Canada, Aug. 1977, pp. 128-134.

18. Manola, F. "On Relating the CODASYL Database Languages and the ANSI/SPARC Framework". Proceedings, Fourth International Conference on Very Large Data Bases, Berlin, West Germany, Sept. 1978, p. 132.

19. Manola, F. "A Review of the 1978 CODASYL Database Specifications". Proceedings, Fourth International Conference on Very Large Data Bases, Berlin, West Germany, Sept. 1978, pp. 232-242.

20. Shneiderman, B. "Improving the Human Factors Aspect of Database Interactions". Unnumbered IFSM Technical Report, University of Maryland, Jan. 1978

21. Smith, D. C. P. "Conversion and the CODASYL Framework". Proceedings, Fourth International Conference on Very Large Data Bases, Berlin, West Germany, Sept. 1978, pp.133-134.

22. Thomas, J. C. "Some Psychological Issues in Data Base Management". Proceedings, Third International Conference on Very Large Data Bases, Tokyo, Japan, Oct. 1978, pp. 169-184.

23. Tsichritzis, D. and Klug, A. "The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems". AFIPS Press, Montvale, N.J., 1977.

DISTRIBUTION LIST

Department of the Navy - Office of Naval Research

Data Base Management Systems Project

Defense Documentation Center
(12 copies)
Cameron Station
Alexandria, VA 22314

Office of Naval Research
Code 102IP
Arlington, Virginia 22217

Office of Naval Research
Branch Office, Chicago
536 South Clark Street
Chicago, IL 60605

New York Area Office

715 Broadway - 5th Floor
New York, NY 10003

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
(Code RD-1)
Washington, DC 20380

Office of Naval Research
Code 458
Arlington, VA 22217

Office of Naval Research
(2 copies)
Information Systems Program
Code 437
Arlington, VA 22217

Office of Naval Research
Branch Office
495 Summer Street
Boston, MA 02210

Office of Naval Research
Branch Office, Pasadena
1030 East Green Street
Pasadena, CA 91106

Naval Research Laboratory
(6 copies)
Technical Information Division
Code 2627
Washington, DC 20375

Office of Naval Research
Code 455
Arlington, VA 22217

Naval Electronics Laboratory Center
Advanced Software Technology Division
Code 5200
San Diego, CA 92152

Mr. E. H. Gleissner
Naval Ship Research and
Development Center
Computation & Mathematics Dept.
Bethesda, MD 20084

Captain Grace M. Hopper
NAICOM/MIS Planning Branch
(OP-916D)
Office of Chief of Naval Operations
Washington, DC 20350

Mr. Kim B. Thompson
Technical Director
Information Systems Division
(OP-911G)
Office of Chief of Naval Operations
Washington, DC 20350

Bureau of Library and
Information Science Research
Rutgers - The State University
189 College Avenue
New Brunswick, NJ 08903
Attn: Dr. Henry Voos

Professor Omar Wing
Columbia University
in the City of New York
Dept. of Electrical Engineering
and Computer Science
New York, NY 10027

Defense Mapping Agency
Topographic Center
ATTN: Advanced Technology
Division
Code 41300 (Mr. W. Mullison)
6500 Brookes Lane
Washington, D.C. 20315

Commander, Naval Sea Systems Command
Department of the Navy
Washington, D.C. 20362
ATTENTION: (PMS30611)

Major J.P. Pennell
Headquarters, Marine Corps
Washington, D.C. 20380
ATTENTION: Code CCA-40

Captain Richard L. Martin, USN
Commanding Officer
USS Francis Marion (LPA-249)
FPO New York 09501

Professor Mike Athans
Massachusetts Institute of Technology
Dept. of Electrical Engineering and
Computer Science
77 Mass. Avenue
Cambridge, MA 02139